A Web-based Petri Nets Application To Teach Music Analysis and Composition

Adriano Baratè

Laboratorio di Informatica Musicale
Dipartimento di Informatica
Università degli Studi di Milano
Via Comelico 39, 20135 Milano, Italy
Email: adriano.barate@unimi.it

***Abstract*—Music Petri nets are a mathematical formalism suitable to express the results of musicological analysis. Being able to infer the structure of a music piece and to represent it through Petri nets is not a trivial task, even for an expert of musicology who is skilled in computational thinking. From this point of view, a computer-based tool can be useful both in the learning phase and in the *a-posteriori* assessment of the achieved results. After providing the theoretical bases about Petri nets, this paper will describe a web application for music education and dissemination, able to integrate and synchronize the results of music analysis with music-related media content within a single on-line framework.**

Keywords–Music; Web; Petri Nets; Analysis; Composition.

I. Introduction

Musicology is the part of the humanities that studies music as a branch of knowledge. Nowadays, the scholarly analysis and research-based study of music can benefit from technology under a number of perspectives, ranging from the technical reproducibility of sound to its preservation through ad-hoc file formats and musical databases, from advanced computing techniques to infer music characteristics to the exploitation of intangible cultural heritage through computer interfaces.

Computational musicology is defined as the study of music with computational modeling and simulation [1]. This discipline is not related only to the use of computers and technological devices, but also to the adoption of statistical and mathematical methods. In this sense, mathematical formalisms aimed at representing music structures and their relationships deserve a particular mention.

The analytical activity that brings to a formal description of music can respond to different goals, including:

- *comprehension*, when the analysis of already existing pieces aims at a deep understanding of their structures and of the music processes the composer had in mind;
- *re-synthesis*, when the final goal is an automatic, computer-driven, or hand-made (but aware) generation of new music materials based on the discovered models.

As explained below, a promising formalism to explain musical processes is the one based on the adoption of Petri nets. Even if Petri nets have been conceived in a context different from information theory, they proved to be a valid means to describe concurrent, asynchronous, and parallel processes, and these features make them suitable to music analysis as well.

Unfortunately, for a student – even trained in music and skilled in computational thinking – the task of carrying out the analysis of a music piece through Petri nets is not trivial: first, such a formalism requires the development of specific analytical skills; moreover, checking the validity of the achieved results is not easy, especially without a way to link the unveiled structures with the original music content. For this reason we have created a set of computer-based tools for the editing, step-by-step execution and synchronized multimedia performance of Petri nets.

In particular, this work focuses on the design and implementation of a web application for music education and dissemination, where the results of music analysis through Petri nets are integrated and synchronized with an advanced media player thanks to the IEEE 1599 technology.

The paper is structured as follows: Section II will introduce the key concepts about Petri nets, Section III will focus on their musical interpretation, Section IV will discuss the issues related to the teaching of Music Petri nets to university students, Section V will describe a web prototype to facilitate learning and result assessment, and finally Section VI will provide some clarifying example.

II. A Short Introduction To Petri Nets

A Petri Net is an abstract and formal model aiming to represent the dynamic behavior of a system with asynchronous and concurrent activities [2]. It can be defined as a directed bipartite graph, in which the *nodes* may represent *transitions* (i.e. events that may occur) and *places* (i.e. conditions). Transitions are linked to places, and vice versa, by directed *arcs*, that describe which places are pre- and/or post-conditions for which transitions. Please note that arcs never run between places or between transitions.

Places in a Petri net may contain a discrete number of marks called *tokens*. Any distribution of tokens over the places will represent a configuration of the net called a *marking*. The upper limit of tokens that a given place can host represents its *capacity*.

A transition in a Petri net may fire if it is enabled, i.e. there are sufficient tokens in all of its input places and there is sufficient room in all of its output places to host newly generated tokens. When the transition fires, it consumes the required input tokens, and creates tokens in its output places. Please note that tokens are not moved from an input to an output place, but consumed in the former and created in the latter.

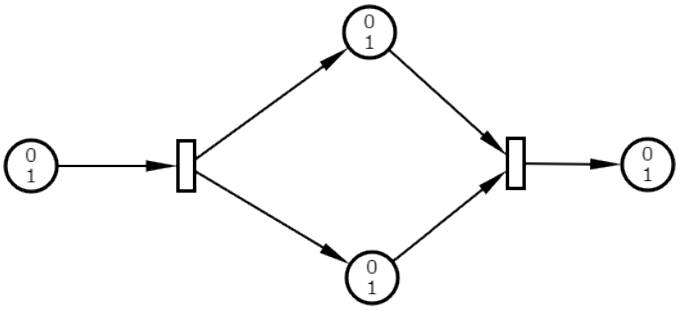

Figure 1. An example of Petri net including 4 places (with no initial marking and capacity set to 1) and 2 transitions.

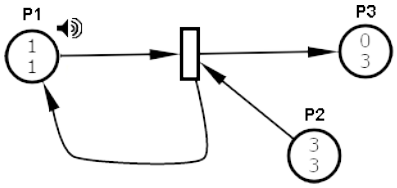

Figure 2. An iterative structure to play the music fragment contained in P1 for $n + 1$ times, with P2 having an initial marking $n = 3$.

Arcs are weighted, and the so-called *arc multiplicity* defines how many tokens should be consumed/created in the corresponding input/output place respectively.

A firing is atomic, in other words it is a single non-interruptible step. Moreover, unless an execution policy is defined, the execution of Petri nets is non-deterministic: when multiple transitions are enabled at the same time, any one of them may fire. Petri nets have an exact mathematical definition of their execution semantics, with a well-developed mathematical theory for process analysis [3].

Thanks to their characteristics, Petri nets are well suited for modeling the behavior of distributed systems. The modeled processes can include choice, iteration, and concurrent execution [4].

Petri nets are usually represented through a graphical formalism, where transitions are symbolized by bars or rectangles, places by circles, and arcs by lines with an arrow head. A conventional way to indicate the current place marking and its capacity is to inscribe in the circle an upper and a lower number respectively. A complete example of graphical representation is shown in Figure 1.

III. Music Petri Nets

A specific extension of Petri nets has been created for music applications. In this implementation, called Music Petri Nets (MPNs), places can be associated to music objects to be played when a token is received, and transitions can contain musical operators that alter the music objects of input places and put these modified objects into output places. A music object may be anything that could have a musical meaning, e.g. a single note, a fragment of music, a control signal, etc. Music operators apply transformational algorithms such as transpositions, inversions, and time stretching. It is worth underlining that in MPNs not all places are necessarily associated to music objects, nor the transitions to musical operators: in this case, these entities are used for mere net evolution, in accordance with their original function in Petri nets. For example, an iterative structure like the one shown in Figure 2 can be adopted to play a music fragment multiple times. This example presents place P1 carrying a musical content and P2 acting as a counter and reserved for net evolution.

MPNs have been applied both to the analysis of already existing pieces of music [5] and to composition and musical expression [6]. In the former case, that is more relevant for our present goals, one factor that clearly influences the analytic power of MPNs is the intrinsic structure of the piece to be described. A composition that mainly contains well-defined music fragments – repeated as they are or after applying some modifications and presenting clear mutual relationships – can be easily mapped onto a limited number of connected structures. The resulting MPN would be compact and effective in the representation of the whole piece structure, as it often happens for counterpoint. Conversely, it would be very difficult to effectively describe a jazz improvisation through this formalism, even after an accurate *a-posteriori* analysis.

IV. Teaching Music Petri Nets

Even if presented to the scientific community in a number of conferences and scientific works, MPNs are deeply rooted in the research activities of the Laboratory of Music Informatics (LIM, Laboratorio di Informatica Musicale) of the University of Milan.

The department of Computer Science of this university offers a degree course in *Music Informatics* that gathers about one hundred freshmen per year. One of the courses that all students have to attend is *Computer Science for Music* (*Informatica applicata alla musica*), where they face technologies and formalisms both from a theoretical and from a practical point of view. In particular, half course (48 hours) is delivered in a computer-equipped classroom. Each lesson lasts approximatively 3 hours, and – after a teacher-led initial part – students are left time for exercises under the supervision of an expert.

In the context of the *Music Informatics* degree, MPNs are seen as a professionalizing subject that may help students investigate the musical processes and provide a formal representation for them. For all we know, this is the only graduate program in the world where MPNs are taught.

This topic is traditionally explained to students in the space of 4 lessons lasting 3 hours each, initially focusing on the theoretical aspects and then increasingly shifting the emphasis towards teacher-guided exercises. At the end of this cycle of lessons the students should be able to:

1) analyze a piece of music and infer its structure in terms of relationships among musical objects;
2) convert this schema to a MPN model;
3) use the mentioned MPN model to create new music pieces, altering the model itself in different ways, e.g. modifying its initial marking or topological characteristics.

The first lessons focus on how traditional Petri nets works, so the basic structures can be presented with no reference to their meaning from a musical perspective. An example of these structure-oriented exercises is shown in Figure 3. After some clarifying examples and assignments, ad-hoc music excerpts can be analyzed (see Figure 4). In the last lessons, students

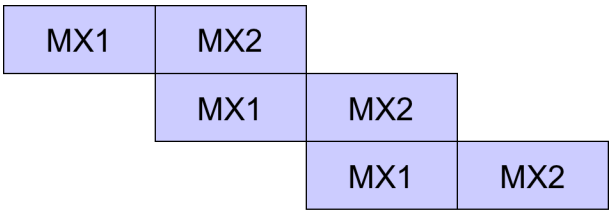

Figure 3. An exercise with a structure presented as an abstract schema.

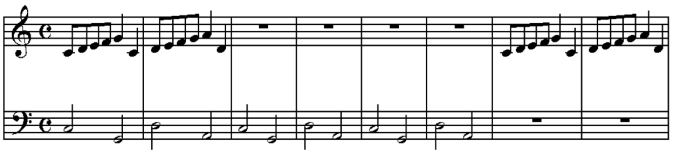

Figure 4. An exercise with a music piece to be analyzed and represented as a MPN.

are invited to use the learned concepts to create new music structures.

The homework assignment for the final exam is to analyze a music excerpt and to provide a MPN formalization of it. There are no constraints about the scores to analyze: music works may belong to any genre and come from any culture, geographical area, and historical period. Moreover, analysis can occur at different degrees of abstraction and detail: as a matter of fact, some students focus on the macro-structural analysis of a complete music piece, other students on a limited number of measures but achieving a very high degree of detail.

In order to foster students' comprehension of this non-trivial subject and to facilitate their tasks, a software tool called *ScoreSynth* – aiming at the representation and step-by-step execution of MPNs – has been released. *ScoreSynth* allows to draw a Petri net of arbitrary complexity, to fix its initial marking, to assign musical fragments to places, and to let the net evolve evaluating its step-by-step behavior, finally achieving the generation (or the reconstruction) of a music score. The interface of *ScoreSynth* is shown in Figure 5.

The student satisfaction towards this free application is quite high, nevertheless *ScoreSynth* presents a number of known limitations:

- it produces a score from a logical point of view, but it cannot perform it;
- the resulting Petri net is not integrated with graphical (score) and audio (performance) contents;
- it requires a specific training to be used;
- consequently, it is mainly an editor rather than a tool to show the expressive power of MPNs to non-experts;
- it works off line, running on a client system;
- it has been developed only for Microsoft Windows™.

In order to solve the mentioned issues, the idea was to shift towards the web-based solution presented in the next section.

V. A Web-based Approach

Before introducing the web-based approach to the visualization, editing, and execution of MPNs, it is worth recalling that the LIM lab has recently worked on an international standard called IEEE 1599, a format that aims to provide a comprehensive description of a music piece in all its aspects. IEEE 1599 is an XML-based standard that can embed multiple and heterogeneous descriptions of a single music piece, all mutually synchronized: metadata, score symbols, graphical content, audio content, and even structural information. Further details are provided in the official documentation [7] and in ad-hoc scientific publications [8]. Moreover, this format has proved to be very effective in a number of music-oriented educational contexts, as discussed in [9].

A detailed discussion about IEEE 1599 would shift the focus from the aims of the present work. For our purposes, it is sufficient to recall that an IEEE 1599 document could host one or more Petri nets coming from a multi-level analysis, link the discovered music objects to music symbols and to their multimedia representations encoded in the document itself, and finally allow a synchronized visualization of all these contents.

Since a web player for IEEE 1599 is already available on the web, the idea was to augment its functionalities by adding the possibility to view and interact with Petri nets. The original web player is integrated in the EMIPIU[1] portal, whose Music Box section demonstrates the potentialities of IEEE 1599 through a number of significant and heterogeneous musical examples. This web application is fully compliant with W3C standards and independent from the hardware and software characteristics of the local system in use. The portal includes project details, official documentation and a community area to exchange opinions, share materials and request clarifications on technical issues.

As mentioned above, for the purposes of this work the most interesting section is the Music Box area, containing the media player that implements advanced navigation and

[1]EMIPIU (Enriched Music Interactive Platform for Internet User) is a publicly-funded international scientific cooperation carried out by the Laboratorio di Informatica Musicale (LIM) - Università degli Studi di Milano and the Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM) - Université Montpellier.

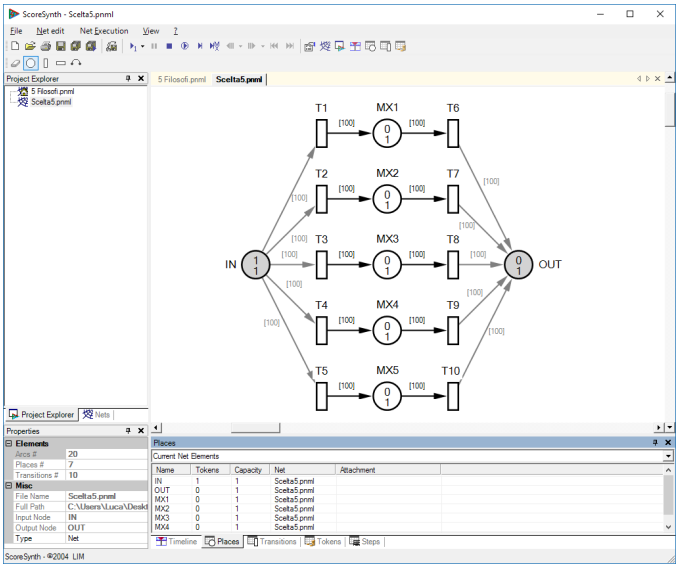

Figure 5. The interface of *ScoreSynth* for Microsoft Windows™.

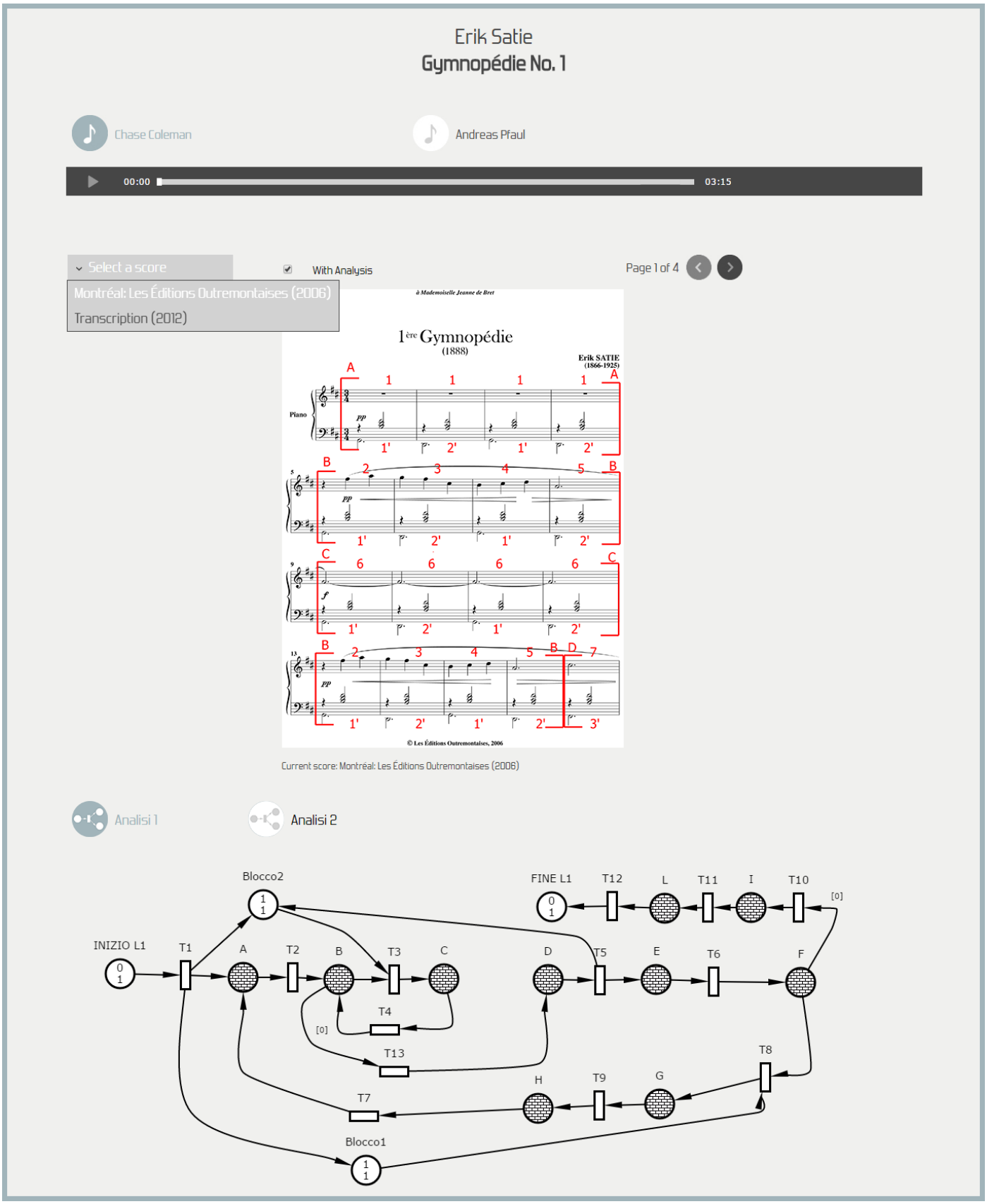

Figure 6. The web interface for MPNs analysis integrated with score and audio.

synchronization of music-related contents. The base URL of the portal is http://emipiu.di.unimi.it/.

Recently, the standard IEEE 1599 player has been integrated with a MPN section. In order to produce the additional materials needed by the player, two new tools have been developed. The first one is used to associate the music objects contained in places to the corresponding fragments of the IEEE 1599 file, whereas the second tool maps the state of the MPN to specific timings of the music piece. In this way, when an audio track is played within the IEEE 1599 player, the corresponding Petri net model can be shown, synchronizing its evolution in terms of markings as the music is advancing, and optionally showing the music fragments associated to places on the score. The entire process (i.e., analysis, generation of the model, and mapping of fragment correspondences in the IEEE 1599 document) helps students check their work, and allows them to experience the achieved results in an interactive environment.

The environment still supports *ScoreSynth* as a way to encode Petri nets, but it provides also high interoperability with other systems thanks to the Petri Net Markup Language (PNML), an XML-based interchange format conceived for the representation of Petri nets [10]. PNML documents can be generated also by *ScoreSynth*, and they can be imported into the IEEE 1599 web player.

This approach intrinsically solves most issues listed in Section IV. First, the audio rendition(s) of the Petri net model is demanded to the IEEE 1599 player, that is also responsible for the operation known as *score following*.[2] Besides, as it regards availability and cross-platform compatibility, the web application can run on any device equipped with an HTML5 browser and connected to the Web. The downside of this approach is that, currently, the application can be used only as a viewer, so the Petri net must be created using *ScoreSynth* or another PNML editor. However, an evolution with editing functions is under development.

Finally, it is worth recalling that a web interface integrated with a media player allows the exploitation of the great amount of work done by students during their final assignments. Until now, results have remained on paper, and it was very difficult to show the descriptive power of MPNs to non-experts.

VI. Examples

The web-based approach described in the previous section supports an integrated experience of music content together with its analysis formalized through MPNs, thus providing the user with the possibility to gain a deep understanding of music processes scaffolded by multimedia.

In this section we will discuss a case study focusing on the *Gymnopédie No. 1* by Erik Satie. The example is available on line at http://satie.lim.di.unimi.it.

Please note that in this context we are not interested in the process that generated a specific musicological analysis. This kind of activity may achieve different results, depending on the music features to analyze, the target degree of abstraction and – obviously – the skills and aims of the expert. For instance, the example we will mention carries two different analyses, which is perfectly consistent with the "multi-instance" approach of IEEE 1599 towards the content of any layer: potentially, multiple scores, multiple audio performances, and multiple analyses as well.

The interface is a variant of the web player for IEEE 1599 documents, with the presence of a Petri net viewer under the score. A drop-down menu allows to select which analysis to follow out of many.[3] A check box enables the visualization of related cues and symbols over the score. The interface is shown in Figure 6.

This viewer supports subnets too, a useful device to make the representation more compact by embedding a complex structure inside. Subnets are formally defined in Petri nets theory, and they essentially provide a compact and readable way to represent the multiple levels of abstraction of a net. In our interface, subnets are graphically rendered through grayed places and they can be double clicked in order to explore their hidden content.

The places that carry musical content are identified through a loudspeaker icon. When the in-going transitions fire, they virtually launch a music fragment. Such places can be clicked to alter their marking and force the performance of the related music content.

As the performance is advancing, the places involved in the execution are highlighted in yellow and their marking changes accordingly.

The *Gymnopédie No. 1* is a good testbed for MPN theory and applications. In fact, even if this composition is far from Renaissance or Baroque contrapuntal style, it presents a number of clearly identifiable music objects, that are literally repeated or slightly variated; compositions of smaller objects form higher-level structures (i.e., musical phrases) that are in turn repeated, and so on.

In conclusion, this example – fully working via web – illustrates the possibility to investigate the structure of a piece of music through multiple analyses, each one composed by multiple layers. The experience of potentially complex and very articulated information is facilitated by a number of features, including the integration with score following and multiple-level subnet exploration. This tool has proved to be effective and even engaging in the learning/teaching activities related to the study of MPNs.

VII. Conclusion

In this work we have presented a new web-based tool that helps *Music Informatics* students understand musicological concepts and music analysis through Petri nets, thus fostering computational thinking skills. Even if models still have to be encoded through *ScoreSynth* or another PNML editor, now it is possible to embed them into IEEE 1599 documents, thus having a single integrated environment to check the analysis and to provide an intuitive multimedia experience.

Since the introduction of this solution in the course of *Computer Science for Music*, the number of errors in the assignments for the final exam has significantly decreased, and student satisfaction has raised, as the author of a musicological analysis can now see (and show to others) the result of his/her efforts.

[2] Score following is the process of listening to a performance and tracking the position in the score.

[3] In this case, the IEEE 1599 document carries only 2 analyses.

As it regards future work, we are planning the development of a complete web-based solution that permits to analyze music pieces, design MPNs models, connect IEEE 1599 documents, and publish the result in an integrated player available on line.

REFERENCES

[1] E. Coutinho, M. Gimenes, J. M. Martins, and E. R. Miranda, "Computational musicology: An artificial life approach," in Artificial intelligence, 2005. epia 2005. portuguese conference on. IEEE, 2005, pp. 85–93.

[2] C. A. Petri, "Introduction to general net theory," in Net theory and applications. Springer, 1980, pp. 1–19.

[3] J. L. Peterson, "Petri nets," ACM Comput. Surv., vol. 9, no. 3, Sep. 1977, pp. 223–252. [Online]. Available: http://doi.acm.org.pros.lib.unimi.it/10.1145/356698.356702

[4] E. Best and R. Devillers, "Sequential and concurrent behaviour in Petri net theory," Theoretical Computer Science, vol. 55, no. 1, 1987, pp. 87–136.

[5] A. Baratè, G. Haus, and L. A. Ludovico, "Music analysis and modeling through Petri nets," in Computer Music Modeling and Retrieval : 3rd International Symposium, CMMR 2005, Pisa, Italy, September 26-28, 2005 : Revised Papers, ser. Lecture Notes in Computer Science, R. Kronland-Martinet, T. Voinier, and S. Ystad, Eds., vol. 3902. Berlin Heidelberg, Germany: Springer-Verlag, 2006, pp. 201–218.

[6] ——, "Real-time music composition through P-timed Petri nets," in ICMC—SMC—2014 Proceedings, Athens 14-20 September 2014, A. Georgaki and G. Kouroupetroglou, Eds., Athens, Greece, 2014, pp. 408–415.

[7] "IEEE recommended practice for defining a commonly acceptable musical application using XML," IEEE Std 1599-2008, Sept 2008, pp. 1–110.

[8] D. L. Baggi and G. M. Haus, Music navigation with symbols and layers: Toward content browsing with IEEE 1599 XML encoding. John Wiley & Sons, 2013.

[9] A. Baratè and L. A. Ludovico, "IEEE 1599 applications for entertainment and education," in Music Navigation with Symbols and Layers: Toward Content Browsing with IEEE 1599 XML Encoding, D. Baggi and G. Haus, Eds. Wiley-IEEE Computer Society Press, 2013, pp. 115–132.

[10] M. Weber and E. Kindler, "The Petri net markup language," in Petri Net Technology for Communication-Based Systems. Springer, 2003, pp. 124–144.